IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Dimpsey et al.**	§	
	§	Group Art Unit: **2185**
Serial No. **10/806,871**	§	
	§	Examiner: **Savla, Arpan P.**
Filed: **March 22, 2004**	§	
	§	
For: **Method and Apparatus for Hardware Assistance for Prefetching a Pointer to a Data Structure Identified by a Prefetch Indicator**	§	

35525

PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on June 11, 2008.

A fee of $510.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447.

No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

This appeal has no related proceedings or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-25

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: 3, 8-10, 13, and 20

Claims withdrawn from consideration but not canceled: None

Claims pending: 1, 2, 4-7, 11, 12, 14-19, and 21-25

Claims allowed: None

Claims rejected: 1, 2, 4-7, 11, 12, 14-19, and 21-25

Claims objected to: None

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 2, 4-7, 11, 12, 14-19, and 21-25

STATUS OF AMENDMENTS

An Amendment after the Final Office Action of April 9, 2008, was not filed. Accordingly, the claims on appeal herein are as amended in the Response to Office Action filed December 21, 2007.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to method, in a data processing system, for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system. Responsive to loading an instruction in the code into a cache, it is determined, by a processor unit, whether a prefetch indicator is associated with the instruction (**2400, 2402, Figure 24**; Specification, page 55, lines 8-11). Responsive to the prefetch indicator being associated with the instruction, a pointer to a data structure identified by the prefetch indicator is selectively prefetched into the cache in the processor (**2406, Figure 24**; Specification, page 56, lines 11-15). The selectively prefetching step includes determining whether outstanding cache misses are present, and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold (**2404, 2406, Figure 24**; Specification, page 55, line 6-page 56, line 1).

B. CLAIM 11 – INDEPENDENT

The subject matter of claim 11 is directed to a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system. The data processing system has determining means (**210**, **214, 228, 300, 950, Figures 2, 3 and 9**; Specification, page 16, lines, 3, 6-7, and 19; page 25, lines 6-8, page 26, line 28, also see page 35, lines 27-28) responsive to loading an instruction in the code into a cache, for determining, by the a processor unit, whether a prefetch indicator is associated with the instruction. The data processing system also has selectively prefetching means (**2400, 2402, Figure 24**; Specification, page 55, lines 8-11), responsive to the prefetch indicator being associated with the instruction, for selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, the selectively prefetching means including means (**102, 104, 106, 210**, **Figures 1, and 2**; Specification, page 13, lines 10-12; page 15, lines, 24-29) for determining whether outstanding cache misses are present, and means (**2404, 2406, Figure 24**; Specification, page 55, line 6-page 56, line 1) for prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.

C. CLAIM 18 – INDEPENDENT

The subject matter of claim 18 is directed to a computer program product in a recordable-type computer readable medium for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system. The computer program product has first instructions, responsive to loading an instruction in the code into a cache, for determining, by a processor unit, whether a prefetch indicator is associated with the instruction (**2400, 2402, Figure 24**; Specification, page 55, lines 8-11), and second instructions, responsive to the prefetch indicator being associated with the instruction, for selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor (**2406, Figure 24**; Specification, page 56, lines 11-15). The second instructions includes first sub-instructions for determining whether outstanding cache misses are present, and second sub-instructions for prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold (**2400, 2406 Figure 24**; Specification, page 55, line 6- 56, line 1).

D. CLAIM 25 – INDEPENDENT

The subject matter of claim 25 is directed to method, in a data processing system, for providing hardware assistance to prefetch data during execution of code by a processor. Responsive to loading an instruction in the code into a cache, it is determined, by a processor unit, whether a prefetch indicator is associated with the instruction (**2400, 2402, Figure 24**; Specification, page 55, lines 8-16). The processor unit is selected from one of an instruction cache, a data cache, and a load/store unit. Responsive to the prefetch indicator being associated with the instruction, a pointer to a data structure identified by the prefetch indicator is selectively prefetched into the cache in the processor (**2406, Figure 24**; Specification, page 56, lines 11-15). The selectively prefetching step includes determining that outstanding cache misses are present; and prefetching the data in response to a determination that a number of the outstanding cache misses is less than a threshold (**2404, 2406, Figure 24**; Specification, page 55, line 6-page 56, line 1). The selectively prefetching step further includes determining to replace cache lines, and prefetching the data in response to a determination that a number of the cache lines chosen to be replaced is greater than a threshold (**2404, 2406, Figure 24**; Specification, page 56, lines 2-7).

E. CLAIM 2 - DEPENDENT

The subject matter of claim 2, which depends from claim 1, recites that the prefetch indicator contains the pointer to the data structure.

F. CLAIM 12 - DEPENDENT

The subject matter of claim 12, which depends from claim 11, specifies that the prefetch indicator contains the pointer to the data structure.

G. CLAIM 19- DEPENDENT

The subject matter of claim 19, which depends from claim 18, specifies that the prefetch indicator contains the pointer to the data structure.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION 1

Claims 1-2, 5-7, 11-12, 15-19, and 22-24 are finally rejected under 35 U.S.C. 103(a) as being obvious over Matsubara et al., U.S. Patent No. 6,381,679 (hereinafter "*Matsubara*") in view of Anonymously Disclosed, "Method for the dynamic prediction of nonsequential memory accesses" (hereinafter "*Anon*") and Ishimi, U.S. Patent No. 5,708,803 (hereinafter "*Ishimi*").

B. GROUND OF REJECTION 2

Claims 4, 14, and 21 are finally rejected under 35 U.S.C. 103(a) as being obvious over *Matsubara* in view of *Anon* and *Ishimi* as applied to claims 1, 11, and 18 above, and further in view of Hooker, U.S. Patent Application Publication No. 2003/0191900 (hereinafter "*Hooker*").

C. GROUND OF REJECTION 3

Claim 25 is finally rejected under 35 U.S.C. 103(a) as being obvious over *Matsubara* in view of *Anon*, *Ishimi*, and *Hooker*.

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1-2, 5-7, 11-12, 15-19, and 22-24)

Claims 1-2, 5-7, 11-12, 15-19, and 22-24 stand finally rejected under 35 U.S.C. 103(a) as being obvious over *Matsubara* in view of *Anon* and *Ishimi*.

A.1. Claims 1-2, 5-7, 11-12, 15-19, and 22-24

In finally rejecting the claims, the Examiner states with respect to independent claims 1 and 18:

> In regards to claim 1 and 18, *Matsubara* discloses:
>
> 4. **As per claims 1 and 18**, Matsubara discloses a method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:
>
> responsive to loading an instruction in the code into a cache, determining, by a processor unit, whether a prefetch indicator is associated with the instruction (col. 5, lines 1-10; col. 6, lines 35-42 and 53-55; col. 7, lines 10-20; Fig. 1; Fig. 2, elements 21 and 22; Fig. 68). *It should be noted that computer program product in claims 18-19 and 21-24 executes the exact same functions as the methods in claims 1-2 and 4-7. Therefore, any references that teach claims 1-2 and 4-7 also teach the corresponding claims 18-19 and 21-24. It should also be noted that the "indication bits (i.e. PF bits)" equaling 1 is analogous to the "prefetch indicator being associated with the instruction" and the "CPU 21" is analogous to the "processor unit." Lastly, it should be noted that the "instruction fetch (IF)"stage is when the instruction in the code is loaded into a cache and the "decoding" stage is when the "determination" is made.*
>
> and responsive to the prefetch indicator being associated with the instruction, selectively prefetching data into the cache in the processor (col. 6, lines 53-55; Fig. 2, elements 21 and 22). *It should be noted that when it is determined that the value of the PF bits is 1, all the data of the line is prefetched to the primary cache.*
>
> Matsubara does not expressly disclose a pointer to a data structure identified by the prefetch indicator;
>
> wherein the selectively prefetching step includes:
>
> determining whether outstanding cache misses are present;
>
> and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.
>
> Anon discloses a pointer to a data structure identified by the prefetch indicator (General Description, 1st paragraph and 4th paragraph; Detailed Description, 1st paragraph). *It should be noted that the "dynamic prefetch pointer" is analogous to the 'pointer to a data structure. "*

Matsubara and Anon are analogous art because they are from the same field of endeavor, that being prefetching memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Anon's dynamic prefetch pointer within Matsubara's information processing system.

The motivation for doing so would have been to improve memory access due to improved memory access prediction and also improve performance due to reducing the time spent waiting for memory accesses to complete (Anon, General Description, 5th paragraph).

The combination of Matsubara/Anon does not expressly disclose wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present;

and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.

Ishimi discloses wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present (col. 13, line 30; Fig. 13, element S4).

and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold (col. 13, lines 30-32; Fig. 13, element S10). *It should be noted that the threshold is equal to 1. Thus, when it is determined there is a cache hit, meaning there are zero outstanding cache misses (i.e. the number of outstanding cache misses is than the threshold of 1), data is prefetched.*

The combination of Matsubara/Anon and lshimi are analogous art because they are from the same field of endeavor, that being prefetching memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Ishimi's fetch mechanism within Matsubara/Anon's information processing system.

The motivation for doing so would have been to provide a data processor capable of processing quickly by lessening the number of abortions even when a branch prediction is preformed (Ishimi, col. 3, lines 3-6).

Therefore, it would have been obvious to combine Matsubara, Anon, and Ishimi for the benefit of obtaining the invention as specified in claims 1 and 18.

Final Office Action dated April 9, 2008, pages 3-5.

Claim 1 on appeal is as follows:

1. A method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:

responsive to loading an instruction in the code into a cache, determining, by a processor unit, whether a prefetch indicator is associated with the instruction; and

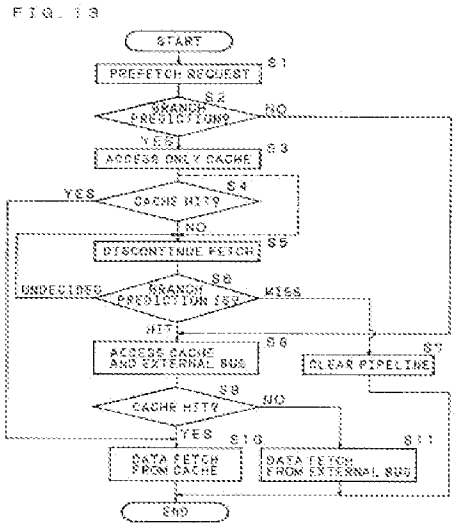
responsive to the prefetch indicator being associated with the instruction, selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present; and

prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). For an invention to be *prima facie* obvious, the prior art must teach or suggest all claim limitations. *In re Royka,* 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974). In this case, the Examiner has not met this burden because all of the recited features of the claims are not found in the cited prior art references as believed by the Examiner. With respect to claim 1, for example, neither *Matsubara* nor *Anon* nor *Ishimi* nor their combination discloses or suggests "determining whether outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold" as recited in claim 1.

In rejecting the claims, the Examiner admits, and Appellants agree, that the combination of *Matsubara* and *Anon* does not teach "determining whether outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold" as recited in claim 1. The Examiner asserts, however, that these features are taught by *Ishimi*, and cites to column 13, lines 30-32 and to Figure 13, element S4 of *Ishimi* as teaching these features. Figure 13 and column 13, lines 30-32 of Ishimi are reproduced below for the convenience of the Board:



Ishimi, Figure 13

> As a result, when the cache hit occurs (step S4), data is fetched from the cache memory in the operand access unit 1 (step S10).

Ishimi, in col. 13, lines 30-32.

Appellants respectfully submit that *Ishii* does not teach or in any way suggest "determining whether outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold" as recited in claim 1. Initially, in order to be able to make a determination that a number of outstanding cache misses is less than a threshold, it would be necessary to make a determination between a number of outstanding cache misses value and a threshold value. *Ishimi*, however, is devoid of any teaching or suggestion in regards to making a determination between a number of outstanding cache misses value and a threshold value.

As depicted in *Ishimi*, Figure 13, reproduced above, at S4, a determination is made as to whether there is a cache hit (or a cache miss). If there is a cache hit, meaning the requested data is presently stored in the cache, the process moves to S10 (Data Fetch From Cache). If there is a cache miss, meaning the requested data is not presently stored in the cache, operand fetch is once aborted at S5 where the result of the branch prediction is not yet determined (see column 13, lines 33-35 of *Ishii*). After the result of the branch prediction is determined, the data is fetched from an external bus as shown at S11. In either scenario, data is fetched. *Ishimi* does not, however, teach or anywhere suggest making a determination that a number of outstanding cache misses is less than a threshold or prefetching data in response to such a determination.

The Examiner states on page 5 of the Final Office Action dated April 9, 2007: "It should be noted that the threshold is equal to 1. Thus, when it is determined there is a cache hit, meaning there are no zero outstanding cache misses (i.e. the number of outstanding cache misses is less than the threshold of 1), data is prefetched." However, *Ishimi* provides no support for and teaches against such a conclusion. *Ishimi* is devoid of any teachings in regards to maintaining a threshold value, and is devoid of any teachings in regards to determining a number of outstanding cache misses. *Ishimi* simply determines if there is a cache hit or a cache miss, and fetches the data from the appropriate location based on that determination. *Ishimi* simply does not teach or

in any way suggest prefetching data in response to a determination that a number of outstanding cache misses is less than a threshold" as recited in claim 1.

Because, as admitted by the Examiner, the combination of *Matsubara* and *Anon* does not teach or suggest "determining whether outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold," as recited in claim 1, and because *Ishimi* also does not teach or suggest such feature as shown above, the combination of *Matsubara*, *Anon*, and *Ishii* considered as a whole, also fails to teach or suggest "determining whether outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold." Accordingly, neither *Matsubara*, nor *Anon*, nor *Ishii* nor their combination discloses or suggests all the features of claim 1 and the Examiner has failed to establish a *prima facie* case of obviousness in rejecting claim 1. Claim 1, accordingly, is not obvious over *Matsubara* in view of *Anon* and *Ishii*, and patentably distinguishes over the cited references in its present form.

Independent claims 11 and 18 recite similar subject matter as claim 1 and also patentably distinguish over *Matsubara* in view of *Anon* and *Ishimi* for similar reasons as discussed above with respect to claim 1. Claims 2, 5-7, 12, 15-17, 19, and 22-24 depend from and further restrict one of independent claims 1, 11 and 18 and also patentably distinguish over *Matsubara* in view of *Anon* and *Ishimi*, at least by virtue of their dependency.

Therefore, claims 1-2, 5-7, 11-12, 15-19, and 22-24 patentably distinguish over *Matsubara* in view of *Anon* and *Ishimi*, and it is respectfully requested that the Board reverse the Examiner's Final Rejection of those claims.

A.2. Claims 2, 12 and 19

The Examiner refers to the fourth paragraph of the General Description in *Anon* as disclosing a prefetch indicator that contains a pointer to a data structure as recited in dependent claims 2, 12 and 19. Appellants respectfully disagree. The publication discloses only a prefetch pointer to predict a probable future memory access. The publication does not disclose or suggest a prefetch indicator (as specifies in the independent claims) that contains a pointer to the data structure identified by the prefetch indicator.

Claims 2, 12 and 19, accordingly, patentably distinguish over the cited references in their own right as well as by virtue of their dependency.

B. GROUND OF REJECTION 2 (Claims 4, 14, and 21)

Claims 4, 14, and 21 stand finally rejected under 35 U.S.C. 103(a) as being obvious over *Matsubara* in view of *Anon* and *Ishimi* as applied to claims 1, 11, and 18 above, and further in view of *Hooker*.

In finally rejecting claims 4, 14 and 21, the Examiner states:

> **As per claims 4 and 21**, the combination of Matsubara/Anon/Ishimi discloses all the limitations of claims 4 and 21 except wherein the selectively prefetching step further includes:
>
> determining whether to replace cache lines;
>
> and prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold.
>
> Hooker discloses wherein the selectively prefetching step further includes: determining whether to replace cache lines (paragraph 0069; Fig. 5, element 536); *It should be noted that the "response buffers" are analogous to the "cache lines."*
>
> and prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold (paragraph 0070; Fig. 5, element 538).
>
> The combination of Matsubara/Anon/Ishimi and Hooker are analogous art because they are from the same field of endeavor, that being prefetching memory systems.
>
> At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Hooker's threshold based prefetch method into Matsubara/Anon/Ishimi's information processing system.
>
> The motivation for doing so would have been to reduce software code size over conventional single-cache line prefetch instructions because fewer prefetch instructions need to be included in the program (Hooker, paragraph 0018). Another motivation for doing so would have been to potentially improve system performance by making more efficient use of the processor bus than the conventional method (Hooker, paragraph 0018). Lastly, another motivation for doing so would have been to potentially improve processing performance by moving data into the microprocessor cache more efficiently than the conventional method by alleviating the problems caused by the fact that a range of core clock to processor bus clock ratios may exist (Hooker, paragraph 0018).
>
> Therefore, it would have been obvious to combine Matsubara, Anon, Ishimi, and Hooker for the benefit of obtaining the invention as specified in claims 4 and 21.
>
> **As per claim 14**, the combination of Matsubara/Anon/Ishimi/Hooker discloses wherein the selectively prefetching means further includes:
>
> means for determining whether to replace cache lines (paragraph 0069; Fig. 5, element 536); *See the citation note for claims 4 and 21 above.*

and means for prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold (paragraph 0070; Fig. 5, element 538).

Final Office Action dated April 9, 2008, pages 9-10.

Claim 4, 14 and 21 depend from and further restrict claims 1, 11 and 18, respectively. *Hooker* is cited as disclosing "wherein the selectively prefetching step further includes: determining whether to replace cache lines." *Hooker* does not supply the deficiencies in *Matsubara*, *Anon* and *Ishii* described above with respect to the independent claims, nor does the Examiner assert otherwise. In particular, *Hooker* also does not disclose or suggest "determining whether outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold" as recited in independent claim 1. Claims 4, 14, and 21, accordingly, patentably distinguish over *Matsubara*, *Anon*, *Ishi* and *Hooker*, at least by virtue of their dependency from independent claims 1, 11, and 18.

Therefore, claims 4, 14 and 21 patentably distinguish over *Matsubara* in view of *Anon*, *Ishimi* and *Hooker*, and it is respectfully requested that the Board reverse the Examiner's Final Rejection of those claims.

C. GROUND OF REJECTION 3 (Claim 25)

Claim 25 stands rejected under 35 U.S.C. 103(a) as being obvious over *Matsubara* in view of *Anon*, *Ishimi*, and *Hooker*.

In finally rejecting claim 25, the Examiner states:

As per claim 25, Matsubara discloses a method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:

responsive to loading an instruction in the code into a cache, determining, by a processor unit, whether a prefetch indicator is associated with the instruction (col. 5, lines 1-10; col. 6, lines 35-42 and 53-55; col. 7, lines 10-20; Fig. 1; Fig. 2, elements 21 and 22; Fig. 6B), wherein the processor unit is selected from one of an instruction cache, data cache, or a load/store unit (Matsubara, col. 6, lines 35-42; col. 7, lines l0-20; Fig. 2, element 21).

and responsive to the prefetch indicator being associated with the instruction, selectively prefetching data into the cache in the processor (col. 6, lines 53-55; Fig. 2, elements 21 and 22).

Matsubara does not expressly disclose a pointer to a data structure identified by the prefetch indicator;

wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present;

and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold, and wherein the selectively prefetching step further includes:

determining whether to replace cache lines;

and prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold.

Anon discloses a pointer to a data structure identified by the prefetch indicator (General Description, 1st paragraph and 4th paragraph; Detailed Description, 1st paragraph). *See the citation notes from the similar limitations in claims 1 and 18 above.*

Matsubara and Anon are analogous art because they are from the same field of endeavor, that being prefetching memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Anon's dynamic prefetch pointer within Matsubara's information processing system.

The motivation for doing so would have been to improve memory access due to improved memory access prediction and also improve performance due to reducing the time spent waiting for memory accesses to complete (Anon, General Description, 5th paragraph).

The combination of Matsubara/Anon does not expressly disclose wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present;

and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold, and wherein the selectively prefetching step further includes:

determining whether to replace cache lines;

and prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold.

Ishimi discloses wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present (col. 13, line 30; Fig. 13, element S4).

and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold (col. 13, lines 30-32; Fig. 13, element S l0).

The combination of Matsubara/Anon and Ishimi are analogous art because they are from the same field of endeavor, that being prefetching memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Ishimi's fetch mechanism within Matsubara/Anon's information processing system.

The motivation for doing so would have been to provide a data processor capable of processing quickly by lessening the number of abortions even when a branch prediction is preformed (Ishimi, col. 3, lines 3-6).

The combination of Matsubara/Anon/Ishimi does not expressly disclose wherein the selectively prefetching step further includes:

determining whether to replace cache lines;

and prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold.

Hooker discloses wherein the selectively prefetching step further includes:

determining whether to replace cache lines (paragraph 0069; Fig. 5, element 536);

and prefetching the in response to a determination that a number of cache lines chosen to replaced is greater than a threshold (paragraph 0070; Fig. 5, element 538).

The combination of Matsubara/Anon/Ishimi and Hooker are analogous art because they are from the same field of endeavor, that being prefetching memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Hooker's threshold based prefetch method into Matsubara/Anon/Ishimi's information processing system.

The motivation for doing so would have been to reduce software code size over conventional single-cache line prefetch instructions because fewer prefetch instructions need to be included in the program (Hooker, paragraph 0018). Another motivation for doing so would have been to potentially improve system performance by making more efficient use of the processor bus than the conventional method (Hooker, paragraph 0018). Lastly, another motivation for doing so would have been to potentially improve processing performance by moving data into the microprocessor cache more efficiently than the conventional method by alleviating the problems caused by the fact that a range of core clock to processor bus clock ratios may exist (Hooker, paragraph 0018).

Therefore, it would have been obvious to combine Matsubara, Anon, Ishimi, and Hooker for the benefit of obtaining the invention as specified in claim 25.

Final Office Action dated April 9, 2008, pages 10-14.

Claim 25 is as follows:

25. A method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:

responsive to loading an instruction in the code into a cache, determining, by a processor unit, whether a prefetch indicator is associated with the instruction, wherein the processor unit is selected from one of an instruction cache, a data cache, and a load/store unit; and

responsive to the prefetch indicator being associated with the instruction, selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, wherein the selectively prefetching step

includes:

determining that outstanding cache misses are present; and

prefetching the data in response to a determination that a number of the outstanding cache misses is less than a threshold, and wherein selectively prefetching step further includes:

determining to replace cache lines; and

prefetching the data in response to a determination that a number of the cache lines chosen to be replaced is greater than a threshold.

For similar reasons as described in detail in Sections A and B above, Appellants respectfully submit that the Examiner has not established a *prima facie* case of obviousness in rejecting claim 25 because neither *Matsubara* nor *Anon* nor *Ishimi* nor *Hooker* nor their combination discloses or suggests "determining that outstanding cache misses are present; and prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold" as recited in claim 25.

Therefore, claim 25 patentably distinguishes over *Matsubara* in view of *Anon*, *Ishimi* and *Hooker*, and it is respectfully requested that the Board reverse the Examiner's Final Rejection of that claim.

/Gerald H. Glanzman/

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777

GG/ma

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:

responsive to loading an instruction in the code into a cache, determining, by a processor unit, whether a prefetch indicator is associated with the instruction; and

responsive to the prefetch indicator being associated with the instruction, selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, wherein the selectively prefetching step includes:

determining whether outstanding cache misses are present; and

prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.

2. The method of claim 1, wherein the prefetch indicator contains the pointer to the data structure.

4. The method of claim 1, wherein the selectively prefetching step further includes:

determining whether to replace cache lines; and

prefetching the data in response to a determination that a number of cache lines chosen to be replaced is greater than a threshold.

5. The method of claim 1, wherein the processor unit is selected from one of an instruction cache, data cache, and a load/store unit.

6. The method of claim 1, wherein the cache is an instruction cache.

7. The method of claim 1, wherein the cache is a data cache.

11. A data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the data processing system comprising:

determining means, responsive to loading an instruction in the code into a cache, for determining, by the a processor unit, whether a prefetch indicator is associated with the instruction; and

selectively prefetching means, responsive to the prefetch indicator being associated with the instruction, for selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, wherein the selectively prefetching means includes:

means for determining whether outstanding cache misses are present; and

means for prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.

12. The data processing system of claim 11, wherein the prefetch indicator contains the pointer to the data structure.

14. The data processing system of claim 11, wherein the selectively prefetching means further includes:

means for determining whether to replace cache lines; and

means for prefetching the data in response to a determination that a number of cache lines chosen to be replaced is greater than a threshold.

15. The data processing system of claim 11, wherein the processor unit is selected from one of an instruction cache, a data cache, and a load/store unit.

16. The data processing system of claim 11, wherein the cache is an instruction cache.

17. The data processing system of claim 11, wherein the cache is a data cache.

18. A computer program product in a recordable-type computer readable medium for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the computer program product comprising:

first instructions, responsive to loading an instruction in the code into a cache, for determining, by the a processor unit, whether a prefetch indicator is associated with the instruction; and

second instructions, responsive to the prefetch indicator being associated with the instruction, for selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, wherein the second instructions includes:

first sub-instructions for determining whether outstanding cache misses are present; and

second sub-instructions for prefetching the data in response to a determination that a number of outstanding cache misses is less than a threshold.

19. The computer program product of claim 18, wherein the prefetch indicator contains the pointer to the data structure.

21. The computer program product of claim 18, wherein the second instructions further includes:

first sub-instructions for determining whether to replace cache lines; and

second sub-instructions for prefetching the data in response to a determination that a number of cache lines chosen to be replaced is greater than a threshold.

22. The computer program product of claim 18, wherein the processor unit is selected from one of an instruction cache, a data cache, and a load/store unit.

23. The computer program product of claim 18, wherein the cache is an instruction cache.

24. The computer program product of claim 18, wherein the cache is a data cache.

25. A method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:

responsive to loading an instruction in the code into a cache, determining, by a processor unit, whether a prefetch indicator is associated with the instruction, wherein the processor unit is

selected from one of an instruction cache, a data cache, and a load/store unit; and

responsive to the prefetch indicator being associated with the instruction, selectively prefetching a pointer to a data structure identified by the prefetch indicator into the cache in the processor, wherein the selectively prefetching step includes:

determining that outstanding cache misses are present; and

prefetching the data in response to a determination that a number of the outstanding cache misses is less than a threshold, and wherein selectively prefetching step further includes:

determining to replace cache lines; and

prefetching the data in response to a determination that a number of the cache lines chosen to be replaced is greater than a threshold.

EVIDENCE APPENDIX

This appeal brief presents no additional evidence.

RELATED PROCEEDINGS APPENDIX

This appeal has no related proceedings.